

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

27A999280
↓
Jc929 U.S. PRO
09/819359
03/28/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 3月31日

出願番号

Application Number:

特願2000-099867

出願人

Applicant (s):

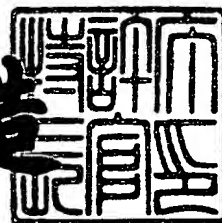
インターナショナル・ビジネス・マシーンズ・コーポレーション

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 9月18日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3074767

【書類名】 特許願

【整理番号】 JA999280

【提出日】 平成12年 3月31日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 1/00

【発明者】

 【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

 【氏名】 羽田 知史

【特許出願人】

 【識別番号】 390009531

 【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

 【識別番号】 100086243

 【弁理士】

 【氏名又は名称】 坂口 博

【代理人】

 【識別番号】 100091568

 【弁理士】

 【氏名又は名称】 市位 嘉宏

【復代理人】

 【識別番号】 100079049

 【弁理士】

 【氏名又は名称】 中島 淳

 【電話番号】 03-3357-5171

【選任した復代理人】

 【識別番号】 100084995

 【弁理士】

【氏名又は名称】 加藤 和詳

【電話番号】 03-3357-5171

【選任した復代理人】

【識別番号】 100085279

【弁理士】

【氏名又は名称】 西元 勝一

【電話番号】 03-3357-5171

【選任した復代理人】

【識別番号】 100099025

【弁理士】

【氏名又は名称】 福田 浩志

【電話番号】 03-3357-5171

【手数料の表示】

【予納台帳番号】 006839

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9304391

【包括委任状番号】 9304392

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ユーザ認証方法、記憶媒体、装置及びシステム

【特許請求の範囲】

【請求項 1】 証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証方法であって、

前記証明者側装置は乱数 a を生成し、暗号 $A = \text{関数 } F(g, a)$ を求め、該暗号 A を検証者側装置に送信し、

前記検証者側装置は乱数 b を生成し、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求め、該暗号 B, X を証明者側装置に送信し、

前記証明者側装置は、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証し、成立すると検証したときに、乱数 c を生成し、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求め、該各暗号を検証者側装置に送信し、

前記検証者側装置は暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の 2 つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証する

ことを特徴とするユーザ認証方法。

【請求項 2】 $(q \mid p - 1)$ の関係を満たす素数 p, q と、位数 q の元を前記整数 g として、公開鍵 v を求めることを特徴とする請求項 1 のユーザ認証方法。

【請求項 3】 前記関数 F は、前記公開鍵 v と秘密鍵 s を用いて

$$v = F(g, -s) = g^{-s} \bmod p$$

の関係を有することを特徴とする請求項 1 のユーザ認証方法。

【請求項 4】 前記証明者側装置は、

$$X = B^a \bmod p$$

の関係が成立したときに乱数 c を生成することを特徴とすることを特徴とする請求項 1 のユーザ認証方法。

【請求項 5】 前記関数 H は、

$$H(a, Y, s) = a + Ys \bmod q$$

の関係を有することを特徴とすることを特徴とする請求項 1 のユーザ認証方法。

【請求項 6】 前記関数 J は、

$$J(v, Y, g, Z) = v^Y g^Z \bmod p$$

の関係を有することを特徴とすることを特徴とする請求項 1 のユーザ認証方法。

【請求項 7】 証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための証明者側装置用プログラムを記録した前記証明者側装置において読み取り可能な記録媒体であって、前記証明者側装置に、

乱数 a を生成させ、暗号 $A =$ 関数 $F(g, a)$ を求めさせ、該暗号 A を検証者側装置に送信させ、

前記検証者側装置からの暗号 B, X を受信させ、

受信された暗号 B, X に基づいて、暗号 $X =$ 関数 $F(B, a)$ の関係の成立性を検証させ、成立すると検証されたときに、乱数 c を生成させ、暗号 $C =$ 関数 $F(g, c)$ 及び暗号 $Y =$ 関数 $F(B, c)$ と、または暗号 $C =$ 関数 $F(A, c)$ 及び暗号 $Y =$ 関数 $F(X, c)$ と、暗号 $Z =$ 関数 $H(a, Y, s)$ とを求めさせ、該各暗号を検証者側装置に送信させる

証明者側装置用プログラムを記録した記録媒体。

【請求項 8】 証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための検証者側装置用プログラムを記録した前記検証者側装置において読み取り可能な記録媒体であって、前記検証者側装置に、

前記証明者側装置からの暗号 A を受信させ、

乱数 b を生成させ、該乱数 b および受信された暗号 A に基づいて、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求め、該暗号 B, X を証明者側装置に送信させ、

前記証明者側装置からの暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを受信させ、

受信された暗号 C, Y, Z に基づいて、暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証させる

検証者側装置用プログラムを記録した記録媒体。

【請求項 9】 証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための証明者側装置用のユーザ認証装置において、

乱数 a を生成しかつ暗号 $A = \text{関数 } F(g, a)$ を求め、求めた暗号 A を検証者側装置に送信する送信手段と、

前記検証者側装置からの暗号 B, X を受信する受信手段と、

受信された暗号 B, X に基づいて、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証する検証手段と、

前記成立すると検証されたときに、乱数 c を生成しかつ、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求める暗号演算手段と、

前記暗号 C, Y, Z を検証者側装置に送信する暗号送信手段と、

を備えたことを特徴とする証明者側装置用のユーザ認証装置。

【請求項 10】 証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定めら

れ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための検証者側装置用のユーザ認証装置において、

前記証明者側装置からの暗号 A を受信する受信手段と、

乱数 b を生成しかつ、該乱数 b および受信された暗号 A に基づいて、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求めて、該暗号 B, X を証明者側装置に送信する送信手段と、

前記証明者側装置からの暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを受信する暗号受信手段と、

受信された暗号 C, Y, Z に基づいて、暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証する検証手段と、

を備えたことを特徴とする検証者側装置用のユーザ認証装置。

【請求項 1 1】 請求項 9 に記載の証明者側装置用のユーザ認証装置と、
複数の請求項 1 0 に記載の検証者側装置用のユーザ認証装置と、
を含むユーザ認証システム。

【請求項 1 2】 証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証システムにおいて、

乱数 a を生成しかつ暗号 $A = \text{関数 } F(g, a)$ を求め、求めた暗号 A を検証者側装置に送信する証明者側装置用送信手段と、

前記証明者側装置からの暗号 A を受信する検証者側装置用受信手段と、

乱数 b を生成し、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求め、該暗号 B, X を証明者側装置に送信する検証者側装置用送信手段と、

前記検証者側装置からの暗号 B, X を受信する証明者側装置用受信手段と、

受信された暗号 B, X に基づいて、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証する証明者側装置用検証手段と、

前記成立すると検証されたときに、乱数 c を生成しかつ、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求める証明者側装置用暗号演算手段と、

前記暗号 C, Y, Z を検証者側装置に送信する証明者側装置用暗号送信手段と、

前記証明者側装置からの暗号 C, Y, Z を受信する検証者側装置用暗号受信手段と、

暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証する検証者側装置用検証手段と、

を含むことを特徴とするユーザ認証システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ネットワークに接続されたコンピュータシステム等の装置におけるユーザ認証方法、ユーザ認証プログラムを記憶した記憶媒体、ユーザ認証装置及びユーザ認証システムにかかり、特に、少なくとも公開鍵が設定された証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証方法、ユーザ認証プログラムを記憶した記憶媒体、ユーザ認証装置及びユーザ認証システムに関する。

【0002】

【従来の技術】

ネットワーク上において、あるユーザが自分の身元を証明するためには、ユーザ認証が必要である。ユーザ認証とは、何らかのプロトコルにより、証明者が自分の身元を検証者に対して証明することであり、電子商取引などの分野において、必須の技術である。例えば、ユーザがサーバに対して、身元を証明したいときは、ユーザが証明者に、サーバが検証者に対応する。また、逆に、サーバがユーザに対して身元を証明したい場合は、サーバが証明者に、ユーザが検証者に対応

する。ユーザ認証は、ユーザとサーバの間に限定されず、任意のコンピュータ間での身元を証明する方法として、幅広く利用されている。最近のユーザ認証では、公開鍵暗号に基づいており、証明者は公開鍵と秘密鍵を所持しており、証明者が公開鍵に対応する秘密鍵を所持することを、なんらかのプロトコルにより、検証者に示すことにより、身元を証明している。

【 0 0 0 3 】

このユーザ認証の代表的な技術として、Schnorrの方法が知られている (C. P. Schnorr, "Efficient Signature Generation by Smart Cards," Journal of Cryptology, Vol.4, No.3, pp.161-174, 1991. 参照)。この技術では、証明者は検証者に対して、公開鍵に対応する秘密鍵を所持することを証明することにより、ユーザ認証を実現している。

【 0 0 0 4 】

従来の技術の一例として、このSchnorrのユーザ認証方法についてその概要を図3を参照して説明する。この方法で用いるシステム・パラメータは、素数 p 、 q (但し、 $q \mid p-1$) と、位数 q の元 $g \in \mathbb{Z}_p$ である。また、証明者の公開鍵は、 v (但し、 $v = g^{-s} \bmod p$) である。証明者の秘密鍵は、 $s \in \mathbb{Z}_q$ である。なお、以下の説明では、証明者及び検証者は、システム・パラメータである素数 p 、 q 、及び元 g も予め取得しているものとし、検証者は証明者の公開鍵 v を予め取得しているものとする。

【 0 0 0 5 】

この方法における検証者と証明者の情報授受は次の通りである。

ステップ1 証明者は乱数 $a \in \mathbb{Z}_q$ を生成し、

$$A = g^a \bmod p$$

を計算し、検証者に送信する。

ステップ2 検証者は乱数 b ($b \in \mathbb{Z}_q$) を生成し、
証明者に送信する。

ステップ3 証明者は

$$c = a + b s \bmod q$$

を計算し、検証者に送信する。

ステップ4 検証者は、

$$A = v^b g^c \bmod p$$

が成立するかどうかを検証し、成立すれば証明者は正当であるとみなす。一方、不成立の場合には、証明者の身元が不当であるとみなし、拒否する。

【0006】

このSchnorrの方法は、離散対数問題に基づく方法の中で、最も効率的である。特に、その通信回数は3回である。しかしながら、その安全性は証明されていない。すなわち、ネットワーク上で、プロトコルが実行される過程で、証明者の秘密鍵 s が露呈する可能性を含んでいる。そこで、証明者と検証者との間の情報授受すなわちユーザ認証（メッセージの送受信等）に対して安全性を評価することが考えられるが、この評価すなわちユーザ認証の安全性について、零知識性に関する技術がよく知られている（S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proofs," Proceedings of 17th symposium on Theory of Computing, pp.291-304, 1985. 参照）。零知識性は、証明者の秘密鍵に関する情報が、一切露呈しないことを意味するものであり、零知識性を満たすことでユーザ認証方法の安全性が保証される。

【0007】

上記のSchnorrの認証方法では、一部を修正することで零知識性を有することになる（A. Fiat and A. Shamir, "How to prove yourself: practical solution to identification and signature problems," Proceedings of Crypto'86, 1980. 参照）。具体的には、検証者による乱数を $b \in \{0, 1\}$ として生成し、プロトコルを $O(\log q)$ 回、シーケンシャルに実行するように修正すれば、零知識性を満たすことが証明されている。すなわち、以下のプロトコルを $O(\log q)$ 回実行し、全ての実行において、検証者が受理するならば、最終的に証明者の身元を認証するものである。

【0008】

〔プロトコル〕

ステップ1：証明者は、乱数 $a \in \mathbb{Z}_q$ を生成し、

$$A = g^a \bmod p$$

を計算し、Aを検証者に送信する。

ステップ2 検証者は乱数 $b \in \{0, 1\}$ を生成し、
証明者に送信する。

ステップ3 証明者は

$$c = a + b s \bmod q$$
を計算し、cを検証者に送信する。

ステップ4 検証者は、

$$A = v^b g^c \bmod p$$
が成立するかどうかを検証し、成立すれば証明者は正当であるとみなす。一方、不成立の場合には、証明者の身元が不当であるとみなし、拒否する。

【0009】

このように、通信回数は $O(\log q)$ に増加するが、零知識性は保証される。上記のSchnorrの認証方法以外にも、零知識性を満たすユーザ認証方法は、従来から数多く知られている。

【0010】

【発明が解決しようとする課題】

しかしながら、従来のユーザ認証方法における零知識性は、証明者と検証者との1対1であることを前提としており、証明者と検証者との間が1対1でプロトコルを実行するときのみに満たされる（図4参照）。すなわち、証明者が複数の検証者と同時にプロトコルを実行する必要がある場合、零知識性が満たされる保証はない（C. Dwork, M. Naor and A. Sahai, "Concurrent Zero-Knowledge," Proc. Of 30th STOC, 1998. 参照）。

【0011】

例えば、インターネットのような非同期型ネットワークでは、複数のコンピュータが同時に通信しており、証明者が複数の検証者と同時にプロトコルを実行する場合がある。WWW (World Wide Web:ワールド・ワイド・ウェブ) では、HTTP (Hypertext Transfer Protocol :WWWサーバーとWWWブラウザやWebブラウ

ザ等が、ファイル等の情報授受に使うプロトコル)のサーバは、接続先である複数のクライアントに対して同時に、自分の身元を証明することが要求される(図5参照)。

【0012】

本発明は、上記事実を考慮して、証明者と検証者が多数の場合であっても零知識性を満たしながら安全にユーザ認証することができるユーザ認証方法、ユーザ認証プログラムを記憶した記憶媒体、ユーザ認証装置及びユーザ認証システムを得ることが目的である。

【0013】

【課題を解決するための手段】

上記目的を達成するために本発明は、証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証方法であって、前記証明者側装置は乱数 a を生成し、暗号 $A = \text{関数 } F(g, a)$ を求め、該暗号 A を検証者側装置に送信し、前記検証者側装置は乱数 b を生成し、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求め、該暗号 B, X を証明者側装置に送信し、前記証明者側装置は、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証し、成立すると検証したときに、乱数 c を生成し、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求め、該各暗号を検証者側装置に送信し、前記検証者側装置は暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証することを特徴とする。

【0014】

$(q | p - 1)$ の関係を満たす素数 p, q と、位数 q の元を前記整数 g として、公開鍵 v を求めることを特徴とすることを特徴とする。

【0015】

前記関数 F は、前記公開鍵 v と秘密鍵 s を用いて

$$v = F(g, -s) = g^{-s} \bmod p$$

の関係を有することを特徴とする。

【0016】

前記証明者側装置は、

$$X = B^a \bmod p$$

の関係が成立したときに乱数 c を生成することを特徴とする。

前記関数 H は、

$$H(a, Y, s) = a + Ys \bmod q$$

の関係を有することを特徴とする。

前記関数 J は、

$$J(v, Y, g, Z) = v^Y g^Z \bmod p$$

の関係を有することを特徴とする。

【0017】

また、他の発明は、証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための証明者側装置用プログラムを記録した前記証明者側装置において読み取り可能な記録媒体であって、前記証明者側装置に、乱数 a を生成させ、暗号 $A = \text{関数 } F(g, a)$ を求めさせ、該暗号 A を検証者側装置に送信させ、前記検証者側装置からの暗号 B 、 X を受信させ、受信された暗号 B 、 X に基づいて、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証させ、成立すると検証されたときに、乱数 c を生成させ、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求めさせ、該各暗号を検証者側装置に送信させる。

【0018】

また、他の発明は、証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の

関係についての正当性を検証するユーザ認証のための検証者側装置用プログラムを記録した前記検証者側装置において読み取り可能な記録媒体であって、前記検証者側装置に、前記証明者側装置からの暗号 A を受信させ、乱数 b を生成させ、該乱数 b および受信された暗号 A に基づいて、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求め、該暗号 B, X を証明者側装置に送信させ、前記証明者側装置からの暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを受信させ、受信された暗号 C, Y, Z に基づいて、暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証させる。

【 0 0 1 9 】

また、他の発明は、証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための証明者側装置用のユーザ認証装置において、乱数 a を生成しかつ暗号 $A = \text{関数 } F(g, a)$ を求め、求めた暗号 A を検証者側装置に送信する送信手段と、前記検証者側装置からの暗号 B, X を受信する受信手段と、受信された暗号 B, X に基づいて、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証する検証手段と、前記成立すると検証されたときに、乱数 c を生成しかつ、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求める暗号演算手段と、前記暗号 C, Y, Z を検証者側装置に送信する暗号送信手段と、を備えたことを特徴とする。

【 0 0 2 0 】

また、他の発明は、証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証のための検証者側装置用のユーザ認

証装置において、前記証明者側装置からの暗号 A を受信する受信手段と、乱数 b を生成しかつ、該乱数 b および受信された暗号 A に基づいて、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求めて、該暗号 B, X を証明者側装置に送信する送信手段と、前記証明者側装置からの暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを受信する暗号受信手段と、受信された暗号 C, Y, Z に基づいて、暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証する検証手段と、を備えたことを特徴とする。

【0021】

また、他の発明のユーザ認証システムは、前記証明者側装置用のユーザ認証装置と、複数の前記検証者側装置用のユーザ認証装置と、を含んでいる。

【0022】

また、他の発明のユーザ認証システムは、証明者側装置の公開鍵 v と秘密鍵 s との関係について予め定めた整数 g を用い、 $v = F(g, -s)$ を満たすべき一方向性関数 F が定められ、証明者側装置と複数の検証者側装置との間で証明者側装置と検証者側装置の関係についての正当性を検証するユーザ認証システムにおいて、乱数 a を生成しかつ暗号 $A = \text{関数 } F(g, a)$ を求め、求めた暗号 A を検証者側装置に送信する証明者側装置用送信手段と、前記証明者側装置からの暗号 A を受信する検証者側装置用受信手段と、乱数 b を生成し、暗号 $B = \text{関数 } F(g, b)$ と暗号 $X = \text{関数 } F(A, b)$ を求め、該暗号 B, X を証明者側装置に送信する検証者側装置用送信手段と、前記検証者側装置からの暗号 B, X を受信する証明者側装置用受信手段と、受信された暗号 B, X に基づいて、暗号 $X = \text{関数 } F(B, a)$ の関係の成立性を検証する証明者側装置用検証手段と、前記成立すると検証されたときに、乱数 c を生成しかつ、暗号 $C = \text{関数 } F(g, c)$ 及び暗号 $Y = \text{関数 } F(B, c)$ と、または暗号 $C = \text{関数 } F(A, c)$ 及び暗号 $Y = \text{関数 } F(X, c)$ と、暗号 $Z = \text{関数 } H(a, Y, s)$ とを求める証明者側装置用暗号演算手段と、前記暗号 C, Y, Z を検証者側装置に送信する証明者側装置用暗号送

信手段と、前記証明者側装置からの暗号 C , Y , Z を受信する検証者側装置用暗号受信手段と、暗号 $Y = \text{関数 } F(C, b)$ 及び暗号 $A = \text{関数 } J(v, Y, g, Z)$ の2つの関係が同時に成立したときに、前記検証者側装置と前記証明者側装置の関係が正当であると検証する検証者側装置用検証手段と、を含むことを特徴とする。

【0023】

【発明の実施の形態】

以下、図面を参照して本発明の実施の形態の一例を詳細に説明する。本実施の形態は、ネットワークにおいて公開鍵 v と秘密鍵 s を用いてユーザ認証する場合に本発明を適用したものである。

【0024】

本発明は、インターネットなどの非同期ネットワークにおいてユーザ認証するためのものである。非同期なネットワークでは、証明者が、複数の検証者に対して、同時に、ユーザ認証のプロトコルを実行されることを要求される可能性がある。つまり、本実施の形態では、1つの証明者に対して、複数の検証者が存在することを想定している。

【0025】

本実施の形態では、暗号化するための関数として次に示す一方向性関数 F を用いている。一方向性関数 F は、2入力、1出力の関数であり、2番目の変域と値域では、それぞれにおいて、加算（+）と乗算（*）の二つの演算が定義されているものとする。

さらに、関数 F は、次の2つの性質を満たす。

すなわち、任意の a , b に関して、

$$\textcircled{1} F(g, a + b) = F(g, a) * F(g, b)$$

$$\textcircled{2} A = F(g, a) \text{ ならば、} F(g, a * b) = F(A, b)$$

の両方が成り立たなければならない。

また、他の暗号化のための関数 H は、次のように表される。関数 H は、3入力、1出力の関数である。

$$H(a, Y, s) = a + Y * s$$

ただし、ここでの加算と乗算は、関数 F の 2 番目の変域で定義されるものである。

さらに、他の暗号化のための関数 J は、4 入力、1 出力の関数であるが関数 F により、次のように表される。

$$J(v, Y, g, Z) = F(v, Y) * F(g, Z)$$

と表される。

【0026】

関数 F の具体例として、離散対数問題に基づく一方向性関数が考えられる。その代表的な例は、素数 p 、 q が $q \mid p-1$ の関係を有し、 $g \in \mathbb{Z}_p$ が位数 q の元であるとしたとき、

$$F(g, a) = g^a \bmod p$$

である。

【0027】

まず、本発明が適用可能な装置を図 2 に示した。ネットワーク 32 には、各々 CPU を少なくとも含むコンピュータにより構成された証明者側装置 10、検証者側装置 40、及び検証者側装置 40 と同様の構成である他の検証者側装置 60 が複数接続されている。このように、本実施の形態では、証明者側装置と、検証者側装置とが 1 対多の接続関係になっている。

【0028】

証明者側装置 10 は、システムパラメータ等を入力するための入力装置 12 を備えており、入力装置 12 は入力に応じた乱数 a を発生する乱数発生器 14 及びメモリ 16 に接続されている。乱数発生器 14 は、メモリ 16 及び乱数 a に基づく暗号 A を求める暗号演算器 18 に接続されている。暗号演算器 18 は、ネットワーク 32 を介して他の装置と通信するためにネットワーク 32 に接続された通信インターフェース（以下、通信 I/F という）30 に接続されている。通信 I/F 30 には、検証器 20 が接続されている。この検証器 20 はメモリ 16 も接続されている。また、検証器 20 は、入力に応じた乱数 c を発生する乱数発生器 22 及び入力信号により後述するプロトコルを中断する中断装置 24 に接続されている。乱数発生器 22 は、乱数 c に基づく暗号 C 、 Y を求める暗号演算器 26 に

接続され、暗号演算器 26 は暗号 C, Y に基づく暗号 Z を求める暗号演算器 28 に接続されている。また、暗号演算器 26、28 は、通信 I/F 30 に接続されると共に、メモリ 16 にも接続されている。

【0029】

検証者側装置 40 は、システムパラメータ等を入力するための入力装置 42 を備えており、入力装置 42 は入力に応じた乱数 b を発生する乱数発生器 44 及びメモリ 46 に接続されている。乱数発生器 44 は、メモリ 46 及び乱数 b に基づく暗号 B, X を求める暗号演算器 48 に接続されている。暗号演算器 18 は、ネットワーク 32 を介して他の装置と通信するために通信 I/F 56 に接続されている。通信 I/F 56 には、検証器 50 が接続されている。この検証器 50 はメモリ 46 も接続されている。また、検証器 20 は、その出力側に、受理装置 52 及び非受理装置 54 に接続されている。

【0030】

なお、他の検証者側装置 60 は、上記検証者側装置 40 と同様の構成であるため、詳細な説明を省略する。以下の説明では、検証者側装置 40 を代表的な構成として、各部の名称を用いるものとする。

【0031】

次に、本実施の形態のプロトコルを説明する。なお、システムパラメータは関数 F_g 、証明者の公開鍵は $v = F(g, -s)$ 、証明者の秘密鍵は s とする。

【0032】

〔プロトコル〕

ステップ 1 : 証明者は、

乱数発生器 14 によって乱数 a を生成し、暗号演算器 18 によって、 $A = F(g, a)$ を計算し、通信 I/F 30 を介して暗号 A を検証者に送信する。

このステップ 1 は、図 1 の証明者側装置 10 の処理 P s 1、及び処理 P s 1 の結果行われる通信 T 1 に相当する。

【0033】

ステップ 2 : 検証者は、

乱数発生器 44 によって乱数 b を生成し、受信した暗号 A を用いて、 $B = F(g$

、 b) および $X = F(A, b)$ を計算し、通信 $I / F 3 0$ を介して暗号 B 、 X を証明者に送信する。

このステップ 2 は、図 1 の検証者側装置 4 0 が通信 $T 1$ を受信した後の処理 $Q s 1$ 、及び処理 $Q s 1$ の結果行われる通信 $T 2$ に相当する。

【0034】

ステップ 3 : 証明者は、

受信した暗号 B 、 X を用いて、検証器 2 0 により、 $X = F(B, a)$ が成立するかを検査し、成立しなければ、検証者が不正を行っていると判断し、中断装置 2 4 によりプロトコルを中断する。一方、成立するならば、乱数 c を生成し、 $C = F(g, c)$ 、及び $Y = F(B, c)$ を計算する。なお、この計算は、 $C = F(A, c)$ 、 $Y = F(X, c)$ でもよい。その後、 $Z = H(a, Y, s)$ すなわち、 $Z = a + Y * s$ を計算し、計算結果である C 、 Y 、 Z を検証者に送信する。

このステップ 3 は、図 1 の証明者側装置 1 0 が通信 $T 2$ を受信した後の処理 $P s 2$ 、及び処理 $P s 2$ の検証器 2 0 による検証が成立した結果行われる通信 $T 3$ に相当する。

【0035】

ステップ 4 : 検証者は、

受信した暗号 C 、 Y 、 Z を用いて、検証器 5 0 により、 $Y = F(c, b)$ および $A = J(v, Y, g, Z)$ 、すなわち $A = F(v, Y) * F(g, Z)$ が成立するかを検査し、両方が成立すれば証明者の身元を受理する（受理装置 5 2 が作動する）。一方、成立しないときは、証明者の身元を拒否する（非受理装置 5 4 が作動する）。

このステップ 4 は、図 1 の検証者側装置 4 0 が通信 $T 3$ を受信した後の処理 $Q s 2$ に相当する。

【0036】

上記プロトコルは、証明者用及び検証者用の処理プログラムとして記録媒体としてのフロッピーディスクに実行可能な形式で格納することができる。この場合、各装置に挿抜可能なフロッピーディスクユニット (FDU) を接続して、フロッピーディスクから FDU を介して記録された処理プログラムを実行すればよい。

。また、処理プログラムをコンピュータ内のRAMや他の記憶領域（例えばハードディスク装置）にアクセス可能に格納にして（インストール）して実行するようにしてもよい。また、予めROMに記憶してもよい。また、記録媒体としては、CD-ROM、MD、MO、DVD等のディスクやDAT等の磁気テープがあり、これらを用いるときには、対応する装置としてCD-ROM装置、MD装置、MO装置、DVD装置、DAT装置等を用いればよい。

【0037】

＜具体例＞

次に、上記プロトコルを用いたユーザ認証の具体例を説明する。なお、以下の具体例では、素数 p 、 q （ただし、 $q \mid p-1$ ）と位数 q の元 g をシステムパラメータとしたときに、関数 F として $v = F(g, -s) = g^{-s} \bmod p$ を用いている。すなわち、鍵構成は、上述のSchnorrと同様のものを用いることができる。また、関数 H として $H(a, Y, s) = a + Ys \bmod q$ を用い、関数 J として $J(v, Y, g, Z) = v^Y g^Z \bmod p$ を用いている。

【0038】

〔鍵構成〕

システムパラメータ：素数 p 、 q （ただし、 $q \mid p-1$ ）と位数 q の元 g

証明者の公開鍵： $v = g^{-s} \bmod p$

証明者の秘密鍵： $s \in Z_q$

【0039】

〔プロトコル〕

ステップ1：証明者は、乱数 a を生成し、暗号 A を計算し、検証者に送信する。

$$a \in Z_q \quad \dots (1)$$

$$A = g^a \bmod p \quad \dots (2)$$

すなわち、証明者側装置10では、乱数発生器14がシステムパラメータ q を用いて（1）式に基づく乱数 a を生成する。そして、暗号演算器18は、その乱数 a 及びシステムパラメータ p 、 g を用いて（2）式により暗号 A を求める。この求めた暗号 A は通信 I/F30に出力され、ネットワーク32を介して検証者

側装置 40 へ送信される。

【0040】

ステップ 2：検証者は、乱数 b を生成し暗号 B と X を計算し証明者に送信する。

$$b \in Z_q \quad \dots (3)$$

$$B = g^b \bmod p \quad \dots (4)$$

$$X = A^b \bmod p \quad \dots (5)$$

すなわち、検証者側装置 40 では、通信 I/F 56 を介して暗号演算器 48 に証明者側装置 10 からの暗号 A が入力される。このとき、検証者側装置 40 では、乱数発生器 44 がシステムパラメータ q を用いて (3) 式に基づく乱数 b を生成する。暗号演算器 48 は、その乱数 b 及び受信した暗号 A を用いて (4)、(5) 式により暗号 B と暗号 X を求める。この求めた暗号 B 、 X は通信 I/F 56 に出力され、ネットワーク 32 を介して証明者側装置 10 へ送信される。

【0041】

ステップ 3：証明者は、暗号 B 、 X を用い、次の (6) 式が成立するかを検査し、成立しなければ、検証者が不正を行っていると判断し、プロトコルを中断する。一方、成立するならば、乱数 c を生成し、暗号 C 、 Y を求める。その後、暗号 Z を求め、暗号 C 、 Y 、 Z を検証者に送信する。

$$X = B^a \bmod p \quad \dots (6)$$

$$c \in Z_q \quad \dots (7)$$

$$C = g^c \bmod p \quad \dots (8)$$

$$Y = B^c \bmod p \quad \dots (9)$$

$$\text{または } C = A^c \bmod p \quad \dots (10)$$

$$Y = X^c \bmod p \quad \dots (11)$$

$$Z = a + Ys \bmod q \quad \dots (12)$$

すなわち、証明者側装置 10 では、通信 I/F 30 を介して検証器 20 に検証者側装置 40 からの暗号 B 、 X が入力される。検証器 20 は、暗号 B 、 X 及びメモリ 16 に記憶したシステムパラメータを用いて上記 (6) 式に従って暗号 B 、 X を検証する。

検証器 20 は、(6) 式が非成立の検証結果である場合、中断装置 24 へ信号

を出力し、プロトコルを中断させる。一方、(6)式が成立する検証結果の場合、検証器20は、乱数発生器22へ信号出力して、乱数発生器44においてシステムパラメータ q に基づく乱数 c を生成させる。この乱数 c は、暗号演算器26へ出力され、暗号演算器26は、その乱数 c 、受信した暗号 B 及びシステムパラメータ p 、 g を用いて上記(8)、(9)式または(10)、(11)に従って暗号 C と Y を求める。次に、求めた暗号 Y 、乱数 a 、秘密鍵 s 、システムパラメータ q を用いて(12)に従って暗号 Z を求める。これら求めた暗号 C 、 Y 、 Z は通信I/F30に出力され、ネットワーク32を介して検証者側装置40へ送信される。

【0042】

ステップ4：検証者は、次の(12)、(13)式が成立するかを検査し、両方が成立すれば証明者の身元を受理する。さもなくば、証明者の身元を拒否する。

$$Y = C^b \bmod p \quad \dots (13)$$

$$A = v^Y g^Z \bmod p \quad \dots (14)$$

すなわち、検証者側装置40では、通信I/F56を介して検証器50に証明者側装置10からの暗号 C 、 Y 、 Z が入力される。検証器50は、暗号 C 、 Y 、 Z 及びメモリ46に記憶したシステムパラメータを用いて上記(13)式と、(14)式に従って暗号 C 、 Y 、 Z を検証する。

検証器50は、(12)、(13)式が非成立の検証結果である場合には、非受理装置54を作動させ、証明者の身元を拒否する。一方、検証器50は、上記式が成立する検証結果の場合には、受理装置52を作動させ、証明者の身元を受理する。

【0043】

本実施の形態では、証明者と検証者との間で、3回の通信のみでユーザ認証が可能である。そして、その通信量は、素数 p 、 q に寄与する。本実施の形態では、通信T1時に暗号 A による $|p|$ 、通信T2時に暗号 B 、 X による $2|p|$ 、通信T3時に暗号 C 、 Y 、 Z による $2|p|$ と $|q|$ である(図1参照)。このため、 $5|p| + |q|$ のわずかな通信量でよいことになる。また、計算量については、上記各式から理解されるように、べき乗の演算負荷の寄与が大きい。こ

のべき乗の回数は、6回でよく、効率的なプロトコルとなることが理解されよう。

【0044】

上記では、証明者と単一の検証者（一人の検証者）との間を例にして説明したが、インターネット等の非同期ネットワークにおいては、多数の検証者に対して証明者の身元確認が必要である。本実施の形態では、各検証者が通信 T1～通信 T3（図1参照）の間の何れの通信状態であっても、その途中の暗号 A, B, C, X, Y, Z から秘匿すべき秘密鍵が傍受されるまたは解析されることはなく、秘匿性を維持することができる。このことは、以下で詳細に説明する。従って、多数の検証者に対して証明者が略同時や逐次に認証する場合でも、多数の検証者の各々に対して確実にユーザ認証することができる。これによって、インターネット等の非同期ネットワークを介して多数の検証者に対して証明者の身元を確認させるときに、安全にユーザ認証することができる。

【0045】

なお、上記具体例は一方方向性関数 F の具体例として Z_p 上のべき乗演算を用いている。いわゆる、離散対数問題に基づく一方方向性関数である。しかし、本発明はこれに限定されるものではない。このほかにも、N を合成数としたとき、 Z_N 上の離散対数問題であってもよいし、また、楕円曲線上の離散対数問題であってもよい。

【0046】

〔プロトコルの有効性〕

次に、本実施の形態のプロトコルの有効性を説明する。すなわち、本実施の形態のプロトコルが非同期ネットワークに適用された場合においても、零知識性を満たすことを、上記説明した＜具体例＞に基づいて説明する。なお、従来の技術の欄に説明したプロトコルでは、非同期ネットワークにおいて零知識性を満たさないことは、すでに知られている（C. Dwork, M. Naor and A. Shai, "Concurrent Zero-Knowledge," Proc. Of 30th STOC, 1998）。

【0047】

非同期ネットワークでは複数の結託する不正な検証者（V1, V2, …,

V_n) は同時に、証明者 P と通信することが可能である。従って、証明者 P と単一の検証者 V との通信を考えるだけでは不十分である。すなわち、証明者 P と複数の検証者 (V_1, V_2, \dots, V_n) との間の通信の零知識性を考える必要がある。

【0048】

そこで、本検証では、複数の結託した不正な検証者 (V_1, V_2, \dots, V_n) が提案するプロトコルに従い証明者 P と通信することによって得ることができる情報は、証明者 P と通信しなくても、得ることができることを証明する。具体的には、任意の不正な検証者 (V_1, V_2, \dots, V_n) に対して、あるアルゴリズム S (シミュレータ) が存在し、 S の出力が、実際の証明者 P と検証者 (V_1, V_2, \dots, V_n) の通信内容の確率分布と一致することを示す。ここでは、「アルゴリズム S は実際の証明者 P と検証者 (V_1, V_2, \dots, V_n) の通信内容をシミュレートする」と表現する。

【0049】

〔検証者の結託〕

一般性を失うことなく、不正な検証者 (V_1, V_2, \dots, V_n) は結託し、以下のように証明者 P と通信すると想定してよい。検証者 (V_1, V_2, \dots, V_n) を検証者 (G_1, G_2, \dots, G_m) にグループ分けする ($m \leq n$)。直観的には、グループ G_i に属する検証者は、グループ G_{i-1} に属する検証者の得た情報に基づいて、証明者 P と通信することを想定している。

【0050】

〔一般化された結託プロトコル〕

入力を証明者 P の公開鍵およびシステムパラメータ (p, q, g, v) とする。

【0051】

ステップ1: 証明者 P は、暗号 $A_1 = g^{a_1}, A_2 = g^{a_2}, \dots, A_n = g^{a_n} \bmod p$ を計算し、暗号 (A_1, A_2, \dots, A_n) を、それぞれ、検証者 (V_1, V_2, \dots, V_n) に送信する。

この時点で検証者が得る情報は、 $VIEW_0 = \{(p, q, g, v), (A_1, A_2, \dots, A_n)\}$ である。

【0052】

ステップ2-1-P: G_1 に属するすべての検証者 V_i は、受信した暗号(A_1, A_2, \dots, A_n)から、乱数 $b_i \in \mathbb{Z}_q$ を生成し、暗号 $B_i (= g^{b_i} \bmod p)$ および暗号 $X_i (= A_i^{b_i} \bmod p)$ を計算し、得られた暗号(B_i, X_i)を証明者 P に送信する。

【0053】

ステップ2-1-V: $V_i \in G_1$ を満たす各 i について、証明者 P は検証式($X_i = B_i^{a_i} \bmod p$)が成り立つか否かを検証する。

もし成り立てば、検証者 V_i に、暗号(C_i, Y_i, Z_i)を送信する。

この時点で、検証者が得る情報は、 $VIEW_1 = VIEW_0 \cup \{(B_i, X_i, C_i, Y_i, Z_i) \mid V_i \in G_1\}$ である。

【0054】

次に、 $2 \leq k \leq n$ に対して、ステップ2-k-P, 2-k-Vを繰り返して処理する。

【0055】

ステップ2-k-P: G_k に属するすべての検証者 V_i は、これまでに得た情報 $VIEW_{k-1}$ から、乱数 $b_i (\in \mathbb{Z}_q)$ を生成し、暗号 $B_i (= g^{b_i} \bmod p)$ 及び暗号 $X_i (= A_i^{b_i} \bmod p)$ を計算し、得られた暗号(B_i, X_i)を証明者 P に送信する。

【0056】

ステップ2-k-V: $V_i \in G_k$ を満たす各 i について、証明者 P は検証式($X_i = B_i^{a_i} \bmod p$)が成り立つか否かを検証する。

もし成り立てば、検証者 V_i に、暗号(C_i, Y_i, Z_i)を送信する。

この時点で、検証者が得る情報は、 $VIEW_k = VIEW_{k-1} \cup \{(B_i, X_i, C_i, Y_i, Z_i) \mid V_i \in G_k\}$ である。

【0057】

以上のことから最終的に結託する検証者が得る情報は

$$VIEW_n = \{(p, q, g, v), \\ (A_1, A_2, \dots, A_n),$$

$$\begin{aligned}
 & (B_1, B_2, \dots, B_n), \\
 & (X_1, X_2, \dots, X_n), \\
 & (C_1, C_2, \dots, C_n), \\
 & (Y_1, Y_2, \dots, Y_n), \\
 & (Z_1, Z_2, \dots, Z_n) \}
 \end{aligned}$$

である。

【0058】

〔結託における計算量的仮定〕

上記ステップ2-k-Vにおいて、各iに対して、

$$X_i = B^{a_i} \bmod p$$

が成り立つためには、検証者V_iは何らかの乱数 $b_i \in \mathbb{Z}_q$ を使って、

$$B_i = g^{b_i} \bmod p \text{ および } X_i = A_i^{b_i} \bmod p$$

を計算しなければならない。つまり、各検証者V_iは乱数 b_i の値を知っていると仮定する。この仮定は、形式的には以下のように述べることができる。

【0059】

〔b-awareness assumption: 以下BAA〕

任意の検証者V_iに対して、上記ステップ2-1-V, 2-2-V, ..., 2-n-Vにおいて、暗号(B_i, X_i)の値だけではなく、乱数 b_i の値も出力するような、別の検証者V_{i'}が存在するとする。

【0060】

〔シミュレーターの構成〕

以下のようにシミュレータSを構成することにより、BAAの下で、零知識性を証明することができる。シミュレータSは検証者(V_1', V_2', \dots, V_n')をサブルーチンとして利用する。これにより、シミュレータSは各乱数 b_i の値を利用することができる。

【0061】

〔シミュレーターのアルゴリズム〕

入力: 公開鍵 v , システムパラメータ p, q, g

出力: $VIEW_n = \{(p, q, g, v),$

$$\begin{aligned} & (A_1, A_2, \dots, A_n), \\ & (B_1, B_2, \dots, B_n), \\ & (X_1, X_2, \dots, X_n), \\ & (C_1, C_2, \dots, C_n), \\ & (Y_1, Y_2, \dots, Y_n), \\ & (Z_1, Z_2, \dots, Z_n) \} \end{aligned}$$

【0062】

ステップ1: 全ての i ($1 \leq i \leq n$) について、乱数 $Y_i \in Z_q$, $Z_i \in Z_q$ を生成し、 $A_i = V^{Y_i} g^{Z_i}$ を計算する。この時点で S がシミュレートした情報は、

$$View_0 = [(p, q, g, v), (A_1, A_2, \dots, A_n)]$$

である。

ステップ2-1-P: G_1 に属するすべての検証者 V_i (V_i') を実行する。

つまり、各 V_i に $VIEW_0$ を入力し、 (B_i, X_i, b_i) を計算する。このとき、 $B_i = g^{b_i} \bmod p$ が成り立つ。

ステップ2-1-V: $Y_i = C_i^{b_i} \bmod p$ を満たす C_i を計算する。この時点で、シミュレータ S がシミュレートした情報は、 $VIEW_1 = VIEW_0 \cup \{(B_i, X_i, C_i, Y_i, Z_i) \mid V_i \in G_1\}$ である。

【0063】

$2 \leq k \leq n$ に対して、ステップ2-k-P, 2-k-V を繰り返す。

【0064】

ステップ2-k-P: G_k に属するすべての検証者 V_i (V_i') を実行する。

つまり、各 V_i に $VIEW_{k-1}$ を入力し、 (B_i, X_i, b_i) を計算する。このとき、 $B_i = g^{b_i} \bmod p$ が成り立つ。

ステップ2-k-V: $Y_i = C_i^{b_i} \bmod p$ を満たす C_i を計算する。この時点で、 S がシミュレートした情報は、 $VIEW_k = VIEW_{k-1} \cup \{(B_i, X_i, C_i, Y_i, Z_i) \mid V_i \in G_k\}$ である

【 0 0 6 5 】

最終的にシミュレートされる通信内容 $VIEW_n$ は、実際の証明者 P と (V_1 , V_2 , \dots , V_n) の通信内容の確立分布と一致する。従って、零知識性が満たされることになる。

【 0 0 6 6 】

【発明の効果】

以上説明したように本発明によれば、証明者側装置と検証者側装置との間で授受される情報から、証明者側装置の秘密鍵が漏洩することがなく、確実にユーザ認証が行える、という効果がある。

特に、インターネットなどの非同期ネットワークを介して、証明者側装置が複数の検証と同時に認証に必要な情報を授受した場合でも、零知識性が満たされるので、そのようなネットワークを介しても、証明者側装置の秘密鍵が漏洩することではなく、確実にユーザ認証を行うことができる、という効果がある。

【図面の簡単な説明】

【図 1】

本発明の実施の形態に係るプロトコルを示すフローチャートである。

【図 2】

本発明の実施の形態に係る証明者側装置と検証者側装置との概略構成を示すブロック図である。

【図 3】

従来のプロトコルを示すフローチャートである。

【図 4】

証明者と検証者とが 1 対 1 で通信することを説明するための説明図である。

【図 5】

証明者と検証者とが 1 対多で通信することを説明するための説明図である。

【符号の説明】

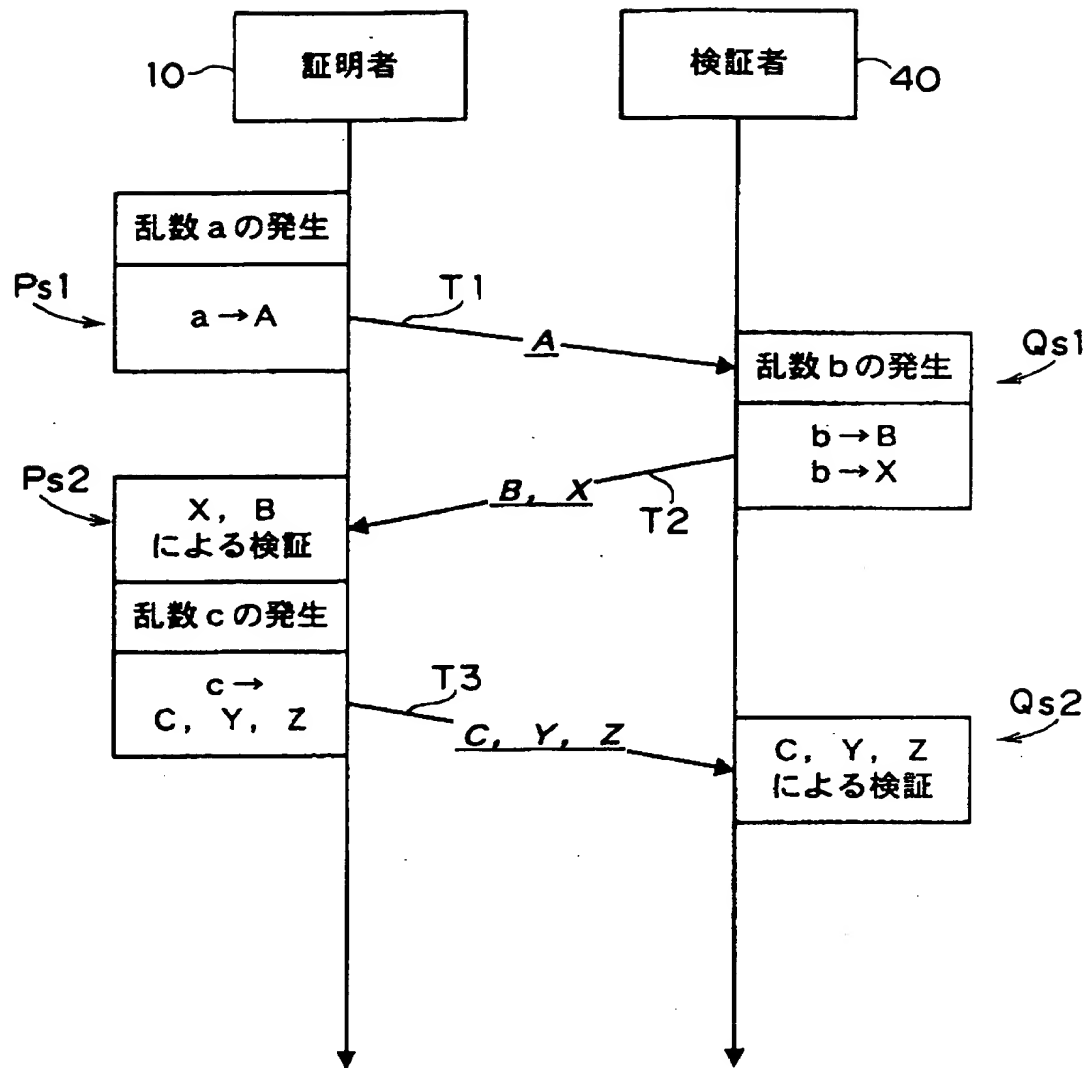
- 1 0 証明者側装置
- 1 2 入力装置

1 4	乱数発生器
1 6	メモリ
1 8	暗号演算器
2 0	検証器
2 2	乱数発生器
2 4	中断装置
2 6	暗号演算器
2 8	暗号演算器
3 2	ネットワーク
4 0	検証者側装置
4 2	入力装置
4 4	乱数発生器
4 6	メモリ
4 8	暗号演算器
5 0	検証器
5 2	受理装置
5 4	非受理装置
6 0	検証者側装置

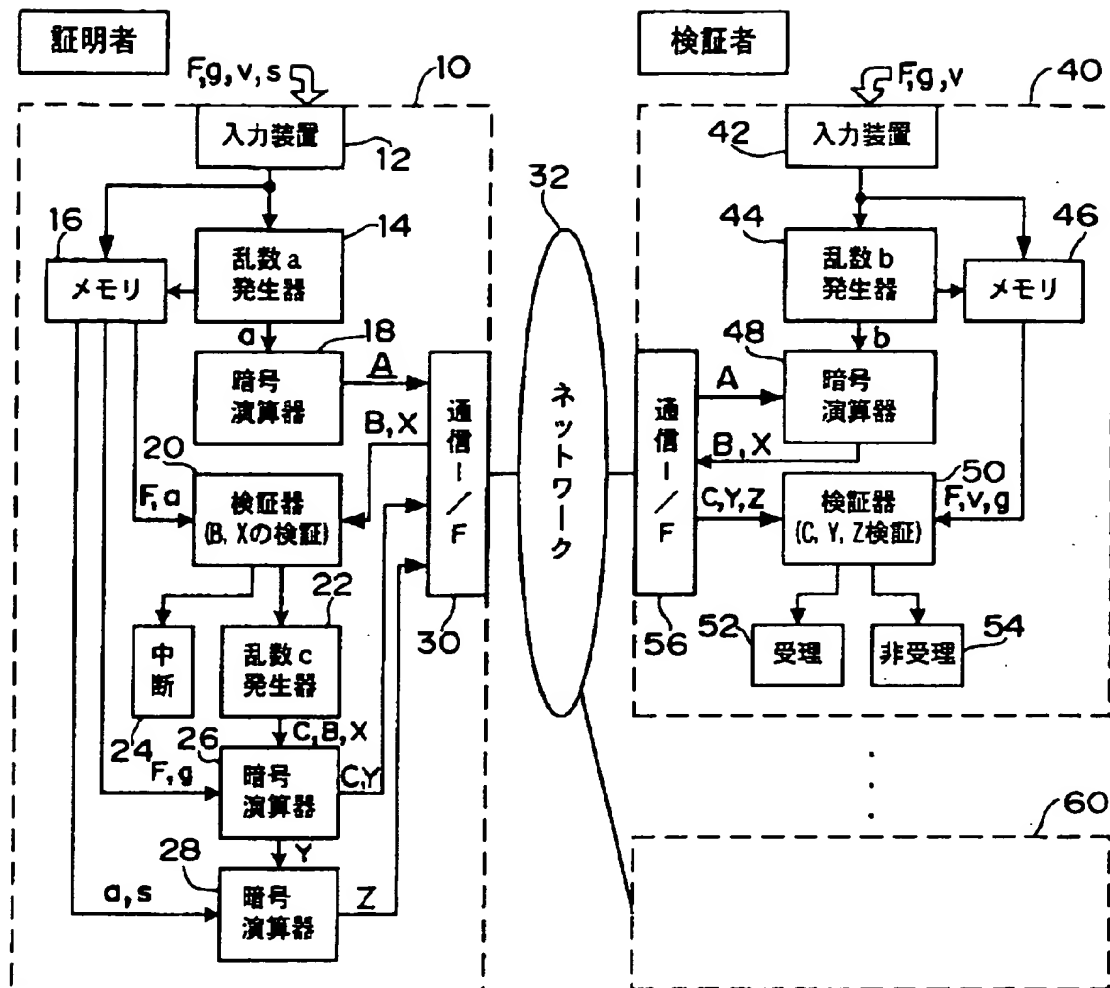
【書類名】

図面

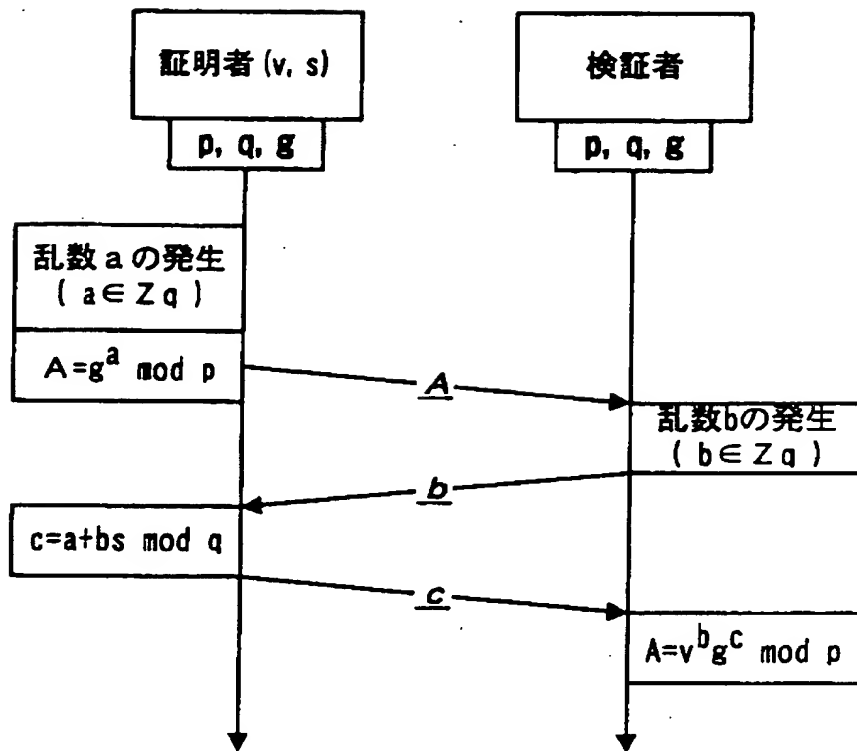
【図 1】



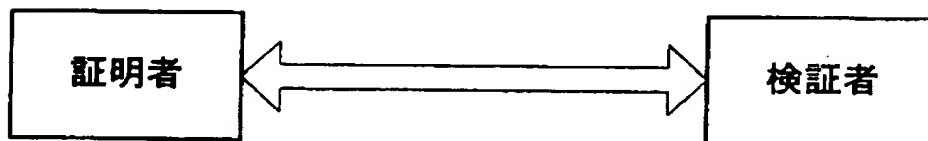
【図 2】



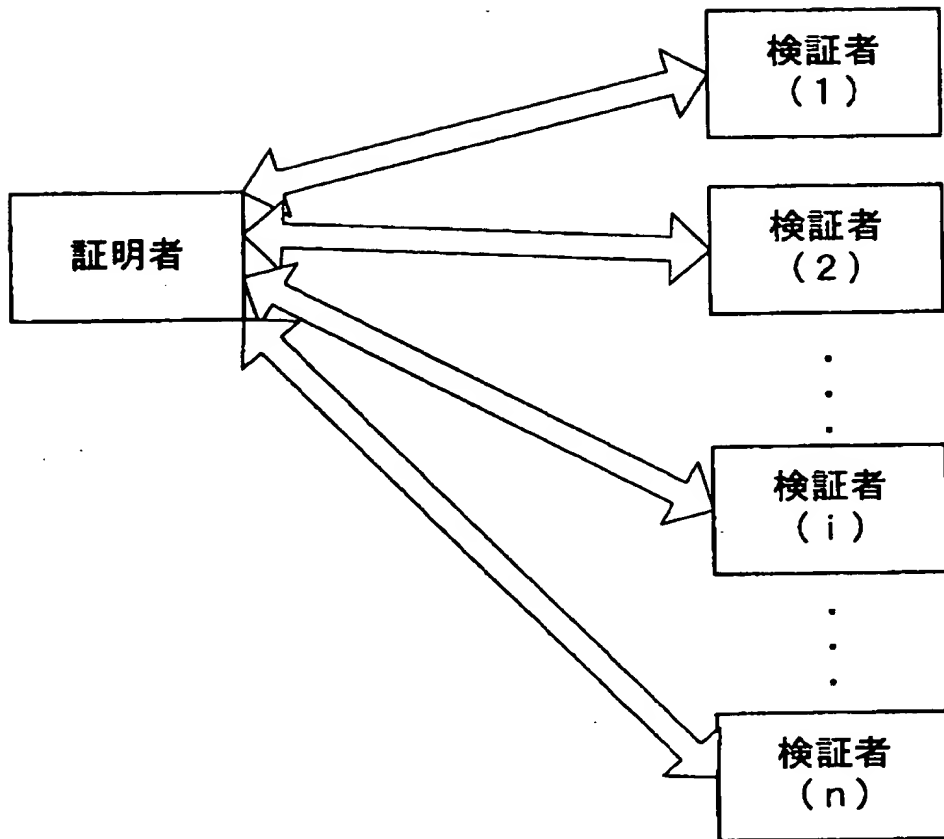
【図 3】



【図 4】



【図 5】



【書類名】 要約書

【要約】

【課題】 証明者と検証者が多数の場合であっても零知識性を満たしながら安全にユーザ認証する。

【解決手段】 プロトコルのステップ1で、証明者10は、乱数 a による $A = F(g, a)$ を計算して検証者に送信する(処理Ps1、通信T1)。ステップ2で、検証者40は、乱数 b による暗号 $B = F(g, b)$ 、 $X = F(A, b)$ を計算して証明者に送信する(処理Qs1、通信T2)。ステップ3で、証明者は、 $X = F(B, a)$ が成立性を検査し、非成立ならばプロトコルを中断し、成立ならば、乱数 c による $C = F(g, c)$ 、 $Y = F(B, c)$ を計算した後、 $Z = H(a, Y, s)$ を計算し、検証者に送信する(処理Ps2、通信T3)。ステップ4で、検証者は、 $Y = F(C, b)$ および $A = J(v, Y, g, Z)$ が成立性を検査し、成立ならば証明者の身元を受理し、非成立ならば証明者の身元を拒否する(処理Qs2)。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2000-099867
受付番号	50000412658
書類名	特許願
担当官	高田 良彦 2319
作成日	平成12年 5月31日

<認定情報・付加情報>

【特許出願人】

【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博

【代理人】

【識別番号】	100091568
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	市位 嘉宏

【復代理人】

【識別番号】	100079049
【住所又は居所】	東京都新宿区新宿4丁目3番17号 HK新宿ビル7階 太陽国際特許事務所
【氏名又は名称】	中島 淳

【選任した復代理人】

【識別番号】	100084995
【住所又は居所】	東京都新宿区新宿4丁目3番17号 HK新宿ビル7階 太陽国際特許事務所
【氏名又は名称】	加藤 和詳

【選任した復代理人】

【識別番号】	100085279
--------	-----------

次頁有

認定・付加情報（続き）

【住所又は居所】	東京都新宿区新宿四丁目3番17号	HK新宿ビル7階	太陽国際特許事務所
【氏名又は名称】	西元	勝一	
【選任した復代理人】			
【識別番号】	100099025		
【住所又は居所】	東京都新宿区新宿4丁目3番17号	HK新宿ビル7階	太陽国際特許事務所
【氏名又は名称】	福田	浩志	

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 1990年10月24日
[変更理由] 新規登録
住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
氏 名 インターナショナル・ビジネス・マシーンズ・コーポレイション

2. 変更年月日 2000年 5月16日
[変更理由] 名称変更
住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーション